Amendments to the Specification

Please replace the paragraph that begins on page 5, line 13 with the following:

As further discussed below various inventive principles and combinations thereof are advantageously employed to facilitate secure communications, including appropriate authorization and authentication of higher layer services or applications, where the authorization and authentication utilize lower layer keying processes. This is facilitated by providing a dynamic key during low level authentication and deriving or generating higher layer, e.g. application level, keys based on the dynamic key, and then providing these application keys as requested by the higher layer applications or services. In this manner, advantageously application level authentication may be accomplished without separately provisioning or configuring a mobile client or network application server, provided each are arranged and operable in accordance with the below described and disclosed principles and concepts.

Please replace paragraph that begins on page 10, line 7 with the following:

The Key Manager 203, 303 or key management utility or entity derives application keys 204, 304 corresponding to the dynamic seed, stores/retrieves 208, 308 these application keys in persistent storage 207, 307, and distributes such keys 205, 305 as required for L3+ services and applications. This is a heretofore unknown and inventive entity, function or component and will be further discussed below. The derivation, distribution, and utilization of application keys is based on layer 2 authentication. This is accomplished using the additional functions or functionality (seed delivery, key acquisition, and key management) at the client and authentication server to allocate and store keys for future use. These functions, namely the Seed Delivery (P/O 201), Key Acquisition 210,310; 212,312; 214,314; 216, 316, and Key Manager functions 203, 303, are inter coupled as shown in FIG. 2 and FIG. 3. A general description of the interaction between these various entities or components will now be provided. Further below is a description of specific instantiation and implementations of this architecture.

Please replace the paragraph that begins on page 12, line 5 with the following:

When an application is launched or initiated, the corresponding application key is delivered or provided, for example via a pull model from the application perspective. On the client side, the application will request the application key corresponding to the application type from the Key Manager 203 that is local, e.g. installed or present on the communication unit. If this key, if present or available, it was generated or derived during the network layer authentication and provided to the Key Manager 203 local to the client device or unit. If no key exists for this application, authentication can not be successfully accomplished and an error will result or occur. If the Key Manager has an application key corresponding to the application making the request, it will provide the key to the application. The application may then use this key directly or as an application seed for generating additional keying material specific to the application. The algorithm used to generate additional application keying material is left to the practitioners choosing given the application specifics and noting that the application client and server will have to use corresponding approaches.

Please replace the paragraph that begins on page 20, line 11 with the following:

Referring to FIG. 6, a ladder diagrams for a SIP application registration using the results of lower layer security keying to support higher layer authentication provisions will be reviewed and discussed. FIG. 6 shows a User Agent Client 601, a client Key Manager 603, and client persistent storage 605. Further depicted is a User Agent Server 607, a RADIUS server 609, a server Key Manager 611, and server persistent storage 613. The interactions or message flow of FIG. 6 illustrates a SIP Registration Scenario that builds upon Layer2 authentication. SIP User Agent Client (UAC) 601 is collocated with the EAP-SIM Client and SIP User Agent Server (UAS) 607 is capable of contacting via the RADIUS server 609 the server Key Manager 611 to retrieve the key. The interactions or message flows shown in FIG. 6 are listed below with their corresponding reference numerals.

Please replace the paragraph that begins on page 18, line 4 with the following:

At 423, a portion of the Key Material, e.g. dynamic seed, generated as a result of the successful EAP SIM based L2 authentication is pushed by the EAP SIM Client Seed Delivery extension to the Client Key Manager 403. Similarly at 425, the identical or functionally identical portion of the material, e.g. dynamic seed, at the server side is pushed by the EAP SIM Server Seed Delivery extension to the Server Key Manger 409. Upon receiving the dynamic seed, The Client Key Manager and Server Key Manager derive Application Keys 427, 429 for various Applications. At 428, 431 the Client Key Manager 403 and Server Key Manager 409 store the Application Keys into the client-side and server-side Persistent Storage 405, 411, respectively, for future use.